

## PATENT ABSTRACTS OF JAPAN

(43) Date of publication of application: 28.01.97

(71) Applicant: FUJIKURA LTD  
(72) Inventor: HIRAGA SEIJI

(57) Abstract:

**SOLUTION:** One character is taken out from a compressed file and the bit of '1' or '1' attached to the branch of a Huffman tree corresponding to the taken-out character is counted in a record position counter. A bit string (code word) outputted to an operation area is outputted to the compression file. Compression to one character is performed, and every time compression for one record is ended, the value (a total bit number inside one record) of the record position counter is added to the compression file. As a result, for the compression file, the positions (terminating positions) of the respective records 1-(n) in a compression data part 91 are added as a record position information part 92 in the order of record numbers as a bit number from the head of the compression file.

Figure 1 is a block diagram of a multi-channel system. A horizontal bar is divided into seven segments. From left to right, the segments are labeled: "ハフマン木情報" (Huffman tree information), "ランレングス" (Run length), "ランレングス" (Run length), "ランレングス" (Run length), "ランレングス" (Run length), "ランレングス" (Run length), and "ランレングス" (Run length). Above the bar, a bracket labeled "94" spans the first two segments, and a bracket labeled "93" spans the next three segments. A bracket labeled "90" points to the entire bar. Below the bar, a bracket labeled "92" spans the first four segments, and a bracket labeled "91" spans the last three segments.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-26902

(43) 公開日 平成9年(1997)1月28日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 12/00  
5/00

識別記号

5 1 1

庁内整理番号

7623-5B

F I

G 0 6 F 12/00  
5/00

5 1 1

H

技術表示箇所

審査請求 未請求 請求項の数 5 O L (全 14 頁)

(21) 出願番号

特願平7-176322

(22) 出願日

平成7年(1995)7月12日

(71) 出願人 000005186

株式会社フジクラ

東京都江東区木場1丁目5番1号

(72) 発明者 平賀 省二

千葉県佐倉市六崎1440番地 株式会社フジ  
クラ佐倉工場内

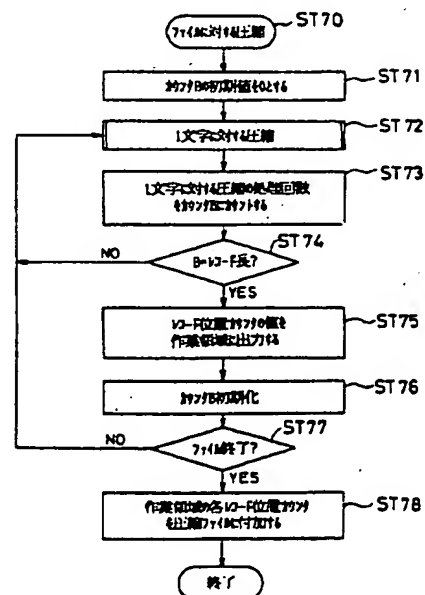
(74) 代理人 弁理士 三好 秀和 (外3名)

(54) 【発明の名称】 ファイル圧縮並びに復元方法

(57) 【要約】

【課題】 特に小型携帯端末機の記憶エリアを節約し、圧縮ファイル中の所定のレコードのみを復元できるようにする。

【解決手段】 複数のレコードで構成された被圧縮ファイルをハフマン法により圧縮する方法において、各レコードの位置情報と各レコード長情報とを圧縮ファイルに付加し、前記レコードの先頭位置情報及びレコード長情報に基づいて圧縮ファイルから所望のレコードのみを復元することを特徴とする。



## 【特許請求の範囲】

【請求項1】 複数の予め定められた長さ（文字列の数）のレコードで構成されたファイルをハフマン法により圧縮する方法において、

圧縮の際に、圧縮前のレコード長（文字列の数）と圧縮後のデータ長とをカウントすることにより、圧縮後の各レコードの位置情報を求め、該各レコードの位置情報を圧縮後のファイルに付加することを特徴とするファイル圧縮方法。

【請求項2】 複数の可変長のレコードで構成されたファイル（10）をハフマン法により圧縮する方法において、前記ファイルは、各レコードの終端にデリミタが挿入されており、圧縮の際に、レコードの文字が前記デリミタに一致した時をもって各レコードの終端と判定して、圧縮後のレコードのデータ長をカウントすることにより、圧縮後の各レコードの位置情報を求め、該各レコードの位置情報を圧縮後のファイルに付加することを特徴とするファイル圧縮方法。

【請求項3】 請求項1の圧縮ファイル又は請求項2の圧縮ファイルから、圧縮後の各レコードの位置情報に基づいて、所望のレコードのみを復元することを特徴とするファイル復元方法。

【請求項4】 被圧縮ファイルの各レコード長の情報を、圧縮後のファイルに付加することを特徴とする請求項1記載のファイル圧縮方法。

【請求項5】 請求項4の圧縮ファイルから、圧縮後の各レコードの位置情報と被圧縮ファイルの各レコード長の情報とに基づいて、所望のレコードのみを復元することを特徴とするファイル復元方法。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、コンピュータやパソコンのデータファイルのデータ圧縮及び復元に関するもので、特にデータ収集・処理・保存をこなすハンディサイズの情報機に関するもので、その補助記憶装置、つまりRAM領域を有効利用することを目的として、ハフマン法と呼ばれるファイル圧縮法を利用したものである。

【0002】

【従来の技術】 ハンディサイズの情報機としての小型携帯端末機は図18に示す構成から成る。図18において、小型携帯端末機1は、各部を制御するCPU1aと、CPU1aの基本的な動作を制御するオペレーティングシステムプログラムが記憶されるROM1bと、対応業務用のアプリケーションプログラムをロードされると共に、該プログラムのワークエリアとして各種レジスタ値が一時記憶されるRAM1cと、内部バスを介してCPU1aから供給される各種データを表示する液晶表示素子等で構成される表示回路1dと、数字入力用のテンキーやファンクションキー等から構成される操作子1

eと、着脱自在に取付けられるメモリカード1fとから構成されている。なお、1gは、例えば、モデム等から構成され、シリアルデータ通信を制御する通信制御回路である。

【0003】 さて、小型携帯端末機1で利用されるデータファイル（システムから読み込み専門に使用される画面情報等のファイルで、固定長レコードを持つランダムアクセスファイル）はパソコンで作成され、ダウンロードにより小型携帯端末機1のRAM1cやメモリカード1fにコピーされて利用される。

【0004】 しかし、RAM1c及びメモリカード1fには容量の限界があり、小型携帯端末機1で大量のデータファイルを扱う大規模なシステムを運用しようとしたとき、データファイルを含むファイルが補助記憶装置に収納出来なくなり、システムの運用が不可能となることも予想される（図17参照）。

【0005】

【発明が解決しようとする課題】 上述の問題を回避するためにデータファイルを圧縮後小型端末機にダウンロードし、小型携帯端末機のシステムで利用するときには復元して使用する方法が考えられるが、圧縮されたデータファイルを元のデータファイルに復元すると、復元されたデータファイルを格納するため、RAMに空き領域が必要である。

【0006】 復元処理時間がかかるといった問題点が出てくる（図16参照）。

【0007】

【課題を解決するための手段】 本発明では、基本的には、ファイルをパソコン上で圧縮し、圧縮されたデータファイルをダウンロードにより小型携帯端末機にコピーして、システムがこれを利用するとき必要なレコードのみをメインメモリ上に復元するものである（図15参照）。

【0008】 上記の目的のために、本発明は、ハフマン法と呼ばれる圧縮アルゴリズムを有効利用することにより実現する。

【0009】 ハフマン法は、後で詳述するが、ハフマン木と呼ばれる符号の変換表によって、文字列中の出現頻度の高い文字をビット長の短い符号語（圧縮された文字）に、出現頻度の低い文字をビット長の長い符号語に変換することで符号化を行なう。よって、生成される符号語はビット単位で構成された長さが一定でない。また、ハフマン木は1つの文字列について一意的に定まり変化しない。また、一意復号可能な符号化を行えるため、各符号語の先頭からであれば復元可能である。

【0010】 従って、基本的には下記（a）、（b）、（c）を行うことにより圧縮ファイル中の任意のレコード（符号語列）を復元できる。

【0011】 （a）圧縮の際に、圧縮後のデータ長をカウントして、圧縮ファイル中の各レコードの位置情報を

求め、圧縮ファイルに追加する。

【0012】(b) 圧縮ファイルから (a) のレコードの位置情報を読みだして、この情報を基に所望のレコードのみを復元する。

【0013】(c) 復元の際に、隣接するレコードの位置情報から所望のレコードの位置、データ長等の情報を求め、その情報に基づいたデータを復元することで所望のレコードを復元する。

【0014】まず、被圧縮ファイルが予め定められた長さ(文字列の数)のレコードで構成されている場合には、下記 (d) を付加して行う。なお、固定長のレコードで構成されている場合は、各レコードが予め定められた長さが一定の場合なので、前記の場合に含まれる。

【0015】(d) 圧縮の際に、レコードの文字数をカウントして、これが前記のレコード長に達した時をもって、各レコードの終端と判定して、(a) の各レコードの位置情報を確定する。

【0016】更に、(c) の代わりに下記 (e)、(f) を付加して行うことにより、復元処理が簡易化できる。

【0017】(e) 被圧縮ファイルの予め定められたレコード長情報を圧縮ファイルに付加する。

【0018】(f) 復元の際に、圧縮ファイルから (e) のレコード長情報を読みだして、そのレコード長の文字列だけを復元することで所望のレコードのみを復元できる。

【0019】次に、被圧縮ファイルが可変長のレコードで構成されている場合には、通常各レコードの終端がデリミタ(区切り文字)で区切られているので、(a)、(b)、(c) に下記 (g) を付加して行う。

【0020】(g) 圧縮の際に、レコードの文字が前記のデリミタに一致した時をもって、そのレコードの終端と判定して (a) の各レコードの位置情報を確定する。

【0021】更に、(c) の代わりに下記 (h) を付加して行うことにより、復元処理が簡易化できる。

【0022】(h) 復元の際に、レコードの文字が前記のデリミタに一致した時をもって、そのレコードの終端と判定して復元を終了することで所望のレコードのみを復元する。

【0023】この発明によれば、パソコン上でハフマン法圧縮アルゴリズムで圧縮されたデータファイルをダウンロードにより小型携帯端末機にコピーして、システムがこれを利用するとき必要なレコードのみをメインメモリ上に復元する。この結果、RAMの空き領域の一部が復元されたデータファイルのために使用されることがない。しかも、レコード長はファイル長より充分短いため、復元処理時間が短い。

【0024】

【発明の実施の形態】以下、図面を参照してこの発明を小型携帯端末機における圧縮ファイルの利用において説

明する。図18にこの発明に係る小型携帯端末機の構成を示すブロック図を示す。この発明は、このような構成からなる小型携帯端末機に適用されるものである。従来、図17に示すように、小型携帯端末機補助記憶装置12にコピーされたデータファイル1のレコードを小型携帯端末機10が読み込む(A1)。あるいは、図16に示すように、圧縮されたデータファイル1を小型携帯端末機に読み込み(B1)、復元した後補助記憶装置12の空き領域へ書き込み(B2)、その復元されたデータファイル1のレコードを小型携帯端末機10が読み込む(B3)。

【0025】本願発明では、図15に示すように、ハフマン法を利用した圧縮法によって圧縮されたデータファイル1の必要なレコードのみを小型携帯端末機10が読み込み(C1)、復元する。復元されたレコードは、メインメモリに置かれる。

【0026】ここで、ハフマン法と呼ばれる圧縮アルゴリズムについて簡単に説明すると、ハフマン法は、ハフマン木と呼ばれる符号の変換表によって、文字列中の出現頻度の高い文字をビット長の短い符号語(圧縮された文字)に、出現頻度の低い文字を長いビット長の符号語に変換することで符号化を行なう。よって、生成される符号語はビット単位で構成された長さが一定でない。

【0027】ハフマン木は、文字列中に出現した文字(葉という)、もしくは節を出現回数が少ないものから節によって束ね、出現回数の合計を節の出現回数とする作業を節が1つになるまで繰り返す(最後の節を根という)ことによって得られる。今、“ADCABAEBAADC”という文字列をハフマン法で符号化すると、構成されるハフマン木は図1のようになる。

【0028】図1において、葉の( )内は、それぞれの文字のデータファイル1中の出現回数である。節、根の( )内は、葉もしくは節の( )内の数字を足し合わせたものである。

【0029】ハフマン木で、根、節、葉をつなぐものを枝と呼ぶ。そして、根、節の各分岐地点において、向かって右下に伸びる枝に1、左下に伸びる枝に0を与え、根から葉に至るまでに辿った枝についていた1と0の並びをビット列としたものがハフマン符号(ハフマン法による符号語)となる。前記の文字列“ADCABAEBAADC”に対するハフマン符号は図2のようになり、符号語の列(圧縮された文字列)は図3のようになる。なお、圧縮後のファイルには、復元時にもハフマン木が利用されるため、ハフマン木の情報が付加される。

【0030】ここで、ハフマン木は、被圧縮ファイル内の全ての文字の出現頻度を調べて作成されるので、1つの文字列について一意的に定まり変化しない。また、一意復号可能な符号化を行なえるため、各符号語の先頭からであれば復元可能である。

【0031】そこで本願発明の基本となる1つの文字を

圧縮する手順とファイルの全体を圧縮する手順とをフローチャートで表して、それぞれ図4と図5とに示す。

【0032】すなわち、図1のハフマン木の枝についた1もしくは0を記録しながら葉（文字）から根までさかのぼり、根に辿り着いたところで記録していた1, 0の列を逆にして、ビット列に変換しながら圧縮ファイルに出力する方法が取られる。ただし、圧縮ファイルの符号語の出力は、バイト単位で行われる。また、ファイルに対する圧縮は、1文字に対する圧縮をファイルの先頭から最後尾までに施せばよい。

【0033】本願発明は、上記のハフマン法による圧縮（符号化）／復元（復号化）アルゴリズムに基づいて行われる。

【0034】また、本願発明では、ハフマン木は1つの文字列について一意的に定まり変化しないことから、各符号語の先頭からであれば、符号語列の任意の符号語から復元可能であることに基づいて、任意のレコードのみを復元するため、前記図4及び図5のフローチャートに下記（a）（b）をつけ加えることが必要となる。

【0035】（a）ハフマン木に基いて圧縮された文字列の総ビット数をカウントする。

【0036】（b）1レコード分の圧縮が終了する度に、符号語の列の総ビット数を配列等の作業領域に記録し、圧縮終了時に圧縮ファイルに付加する。

【0037】すなわち、従来のハフマン法による圧縮ファイルは図9に示すように、符号化された圧縮データ部81にハフマン木情報82が付加されたものである。圧縮データ部81中の各レコード1, 2, …… , n, は構成する符号語がビット単位の長さが一定しないことにより、レコードのランダムアクセスが不可能となる。

【0038】本発明では、復元時に任意のレコードのみを復元するため、上記従来の圧縮ファイルに、図10に示すように圧縮データ部91中の各レコード（1, 2, …… , n, ）の位置がレコード番号順にレコード位置情報部92としてつけ加えられる。

【0039】以下、請求項に対応して実施形態を説明する。

【0040】（実施例1）請求項1に対応した、被圧縮ファイルが複数の予め定められた長さ（文字列の数）のレコードで構成されている場合の、ファイルの圧縮方法の一実施例を説明する。

【0041】まず、図6のフローチャートに示す1文字に対する圧縮がスタートする（ST60）。なお、先だって、被圧縮ファイルの1レコードについてハフマン木を形成しておく必要がある。

【0042】被圧縮ファイルから1文字取り出す（ST61）。ステップST61で取り出した文字に対応するハフマン木の葉を見つける（ST62）。この葉の1つ上の節まで遡り辿った枝についていた0もしくは1を作業領域に出力する（ST63）。ステップST63で出

力した「0」もしくは「1」のビットはレコード位置カウンタでカウントする（ST64）。根に辿り着くまでステップST63及びST64が続けられる（ST65）。かくして作業領域に出力されたビット列（符号語）を圧縮ファイルへ出力する（ST66）。これで1文字に対する圧縮は終了する。

【0043】次いで、ファイルに対する圧縮は、図7に示すフローで行なわれ、ファイルに対する圧縮がスタートする（ST70）。図6のフローで示した1文字に対する圧縮を行い、圧縮の処理回数をカウンタBにカウントする（ST72, ST73）。これに先だってカウンタBの初期値を0とする（ST71）。カウンタBの値がレコード長になるまでステップST72, ST73が続けられる（ST74）。このとき1レコード分の圧縮が終了する度に、レコード位置カウンタの値（1レコード内の総ビット数）を作業領域に出力する（ST75）。そうしてカウンタBを初期化する（ST76）。上記のステップST72～ST76を1ファイルが終了するまで続けられる（ST77）。ステップST75で作業領域に出力していた各レコード位置カウンタの値を圧縮ファイルに付加して（ST78）ファイルに対する圧縮は終了する。この結果、圧縮ファイルは図10に示したように圧縮データ部91中の各レコード（1, 2, …… , n, ）の位置（終端位置）が、圧縮ファイルの先頭からのビット数としてレコード番号順にレコード位置情報部92としてつけ加えられる。

【0044】なお、被圧縮ファイルのレコード長は、圧縮の都度、数値を入力することが適当であるが、常に一定のレコード長である場合には、圧縮のプログラムに入力されていてもよい。

【0045】更に本方法は、被圧縮ファイルの各レコードの長さが異なる場合でも、例えば図7のST74のレコード長をレコード順に切り替えることによって、適用することができる。

【0046】（実施例2）請求項2に対応した、被圧縮ファイルが複数の可変長のレコードで構成されている場合の、ファイルの圧縮方法の一実施例を説明する。

【0047】このような可変長レコードで構成されたファイルは、通常、各レコードの終端がデリミタ（区切り文字）で区切られており、ファイルに対する圧縮は、図8に示すフローで行なわれる。図6のフローで示した1文字に対する圧縮を行い（ST81）、レコード終端に挿入されたデリミタを圧縮する（ST82）まで続ける。このときのレコード位置カウンタの値を作業領域に出力する（ST83）。上記のステップST81～ST83を1ファイルが終了するまで続けられる（ST84）。ステップST83で作業領域に出力していた各レコード位置カウンタの値を圧縮ファイルに付加して（ST85）ファイルに対する圧縮は終了する。この結果、圧縮ファイルは図10に示したように圧縮データ部91

中の各レコード (1, 2, …, n, ) の位置 (終端位置) が、圧縮ファイルの先頭からのビット数としてレコード番号順にレコード位置情報部 92 としてつけ加えられる。

【0048】 (実施例3) 請求項3に対応した、実施例1又は実施例2の圧縮ファイルの復元方法の一実施例を説明する。

【0049】 まず圧縮ファイルから任意のレコードのみを復元する方法について説明する。図10に示した圧縮ファイルの「各レコードの位置」データは、各レコードの終端位置データなので、復元するレコードを  $i$  番目とすると、圧縮ファイルから「 $i-1$  番目のレコードの位置」データを読み込む。

【0050】 このデータは、ファイルの先端から  $i-1$  番目のレコードの終端までの総ビット数で表わされている。図10の圧縮データ部91を拡大すると図11のようになるので、前記のデータの総ビット数を  $(j-1)$

[バイト] +  $k$  [ビット] のバイト単位に変換することができる。

【0051】 次に、ファイルポインタにより圧縮データ部から  $i$  番目のレコードを取り出す方法について説明する。ファイルポインタとは、C言語における論理アドレスのことで、バイト単位でしか移動できないので前記の換算を行っている。図11に示すようにファイルポインタを圧縮データ部91の始端から  $(j-1)$  [バイト] 移動させた後、 $j$  [バイト] 目の  $(k+1)$  [ビット] 目から復元処理を始め、レコード長分だけ処理を継続させれば、 $i$  番目のレコードのみを復元することが出来る。

【0052】 ここで、圧縮ファイルの「 $i$  番目のレコード位置」は、 $i$  番目までのレコードの終端までのデータなので、 $i$  番目のレコードのレコード長は、「 $i$  番目のレコード位置」データを読みこんで、「 $i-1$  番目のレコード位置」データを減算することにより求められる。

【0053】 なお、復元するときも、ハフマン木情報が利用される。

【0054】 次にフローチャートにてより詳細に説明する。

【0055】 まず、1文字に対する復元について説明する。1文字に対する復元は、図12(a)のフローに示すように、圧縮ファイルから1ビット取り出す (STa121)。取出したビットにより図1に示したハフマン木を1つ下の節まで下げる、つまり取出したビットが「0」のときは左下の節、「1」のときは右下の節まで下げる (STa122)。このとき1ビットに対する処理回数をカウンタ  $A'$  にカウントする (STa123)。ステップ STa122 においてハフマン木の葉に辿りつくまで行う (STa124)。葉に対応した文字をメモリに出力して (STa125)、1文字に対する復元は終了する。

【0056】 次に、レコードの復元について説明する。その復元フローは図14に示す。この場合の圧縮ファイルには、レコード位置情報は、例えば  $(i-1)$  番目のレコードの終りが圧縮ファイル頭初からの総ビット数を  $n$  として、 $i$  番目のレコード終りが圧縮ファイル頭始からの総ビット数を  $m$  として圧縮ファイル中のそれぞれ「レコード  $i$  の位置」と「レコード  $i-1$  の位置」に記憶されている。圧縮後の「レコード  $i$ 」のレコード長は  $m-n$  で求められる。

10 【0057】 圧縮ファイルから復元対象レコードの位置情報とレコード長情報 (前記のように復元対象レコードと隣接するレコードの位置情報から換算) を取り出す (ST141)。取り出した  $i$  番目のレコードの位置を読み込むため、図11で説明したように、ファイルポインタを  $(j-1)$  バイト移動させる (ST143)。  $j$  バイト目の先頭から  $(k+1)$  ビット目を取り出し (ST144)、その  $(k+1)$  ビットから1文字に対する復元を開始する (ST145)。1ビットに対する処理回数のカウンタ値  $A'$  が圧縮後のレコード長になった (ST146) とき  $i$  番目レコードの復元は終了する。

【0058】 被圧縮ファイルが可変長のレコードで構成されている場合で、各レコードの終端がデリミタ (区切り文字) で区切られている場合には、レコードの文字が前記のデリミタに一致した時をもって、そのレコードの終端と判定して復元を終了することで所望のレコードのみを復元してもよい。

20 【0059】 (実施例4) 請求項4及び請求項5に対応した、被圧縮ファイルが複数の予め定められた長さ (文字列の数) のレコードで構成されている場合の圧縮及び復元方法の一実施例を説明する。この方法は、被圧縮ファイルの予め定められたレコード長情報が、圧縮ファイルに付加されていることに特徴がある方法であり、前記の実施例の方法に比べて復元方法のプログラムソフトが簡潔化され、その結果、処理の高速化が図れる利点を有する。レコード長情報は、例えば圧縮ファイルのヘッダに付加することが望ましいが、他の場所、例えばレコード位置情報部等に付加してもよい。

30 【0060】 まず、請求項4に対応したファイルの圧縮方法は、実施例1の方法と同様である。次いで、請求項5に対応した圧縮ファイルの復元方法について説明する。

40 【0061】 この場合の1文字に対する復元のフローと、レコードに対する復元のフローをそれぞれ図12(b)と図13に示す。図12(b)において、1文字に対する復元は、圧縮ファイルから1ビット取り出す (STb121)。取出したビットにより図1に示したハフマン木を1つ下の節まで下げる。つまり取出したビットが「0」のときは左下の節、「1」のときは右下の節まで下げる (STb122)。ステップ STb122 においてハフマン木の葉に辿りつくまで行う (ST

b123)。葉に対応した文字をメモリに出力して (STb124)、1文字に対する復元は終了する。図13において、圧縮ファイルから復元対象レコードの位置情報とレコード長を取り出す (ST131)。取り出した i 番目レコードの位置を読み込むため、図11で説明したように、ファイルポインタを (j-1) バイト移動させる (ST133)。圧縮ファイルの圧縮データ部の j バイト目の先頭から (k+1) ビット目を取り出し (ST134)、その (k+1) ビットから1文字に対する復元を開始する (ST135)。このときから復元の処理回数のカウンタを始め (ST136)、カウンタ A の値が圧縮前のレコード長に達した (ST137) ととき i 番目レコードの復元が終了する。

【0062】なお、被圧縮ファイルのレコード長は、圧縮又は復元の都度、数値を入力することが望ましいが、常に一定のレコード長である場合には、圧縮又は復元のプログラムに入力されていてもよい。

【0063】以上の実施例において、圧縮後の各レコードの位置情報は、最初のレコードのデータの始端から各レコードのデータの終端までのビット数で表しているが、他の情報、例えば、各レコードの始端までのビット数、各レコードのデータ長等でも良い。

【0064】-

【発明の効果】以上説明したように、ハフマン木は1つの文字列について一意的に定まり変化しない。また、一意復号可能な符号化を行なえるため、圧縮されたデータは各符号語の先頭からであれば、任意のレコード (符号語列) からの復元が可能である。

【0065】よって、圧縮時に各レコードの位置情報を記録し、圧縮ファイルに付加することにより、任意のレコードを復元できる。

【0066】例えば、パソコン上でハフマン法により圧縮されたデータファイルをダウンロードにより小型携帯端末機にコピーして、これを利用するとき必要なレコードのみをメインメモリ上に復元できる。この結果、小型携帯端末機の補助記憶装置である RAM の空き領域が復元されたデータファイルのために使用されることがない。

【0067】また、ファイル長より充分短いレコードの

みを復元するので、圧縮ファイル全体に対する復元より処理時間が短い。

【図面の簡単な説明】

【図1】圧縮アルゴリズムのハフマン木を示す。

【図2】図1の文字列に対するハフマン符号を示す。

【図3】圧縮された文字列を示す。

【図4】従来の1文字に対する圧縮動作を示すフローチャート図。

【図5】従来のファイルに対する圧縮動作を示すフローチャート図。

【図6】この発明の実施形態による1文字に対する圧縮動作を示すフローチャート図。

【図7】この発明の実施形態によるファイルに対する圧縮動作を示すフローチャート図。

【図8】この発明の他の実施形態によるファイルに対する圧縮動作を示すフローチャート図。

【図9】従来の圧縮ファイル構造の説明図。

【図10】この発明の実施形態による圧縮ファイル構造の説明図。

【図11】この発明の実施形態による復元時に任意のレコードのみを復元する方法の説明図。

【図12】(a)、(b)のいずれも、異なる実施例におけるこの発明の実施形態による1文字に対する復元動作を示すフローチャート図。

【図13】この発明の実施形態によるレコードに対する復元動作を示すフローチャート図。

【図14】この発明の他の実施形態によるレコードに対する復元動作を示すフローチャート図。

【図15】この発明のデータファイルの利用方法の説明図。

【図16】従来のデータファイルの利用方法の説明図。

【図17】従来のデータファイルの利用方法の説明図。

【図18】小型携帯端末機1の構成例を示すブロック図。

【符号の説明】

1 小型携帯端末機

10 小型携帯端末機のシステム

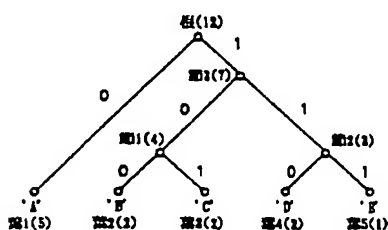
12 小型携帯端末機補助記憶装置 (RAM)

80, 90 圧縮ファイル

【図1】

【図2】

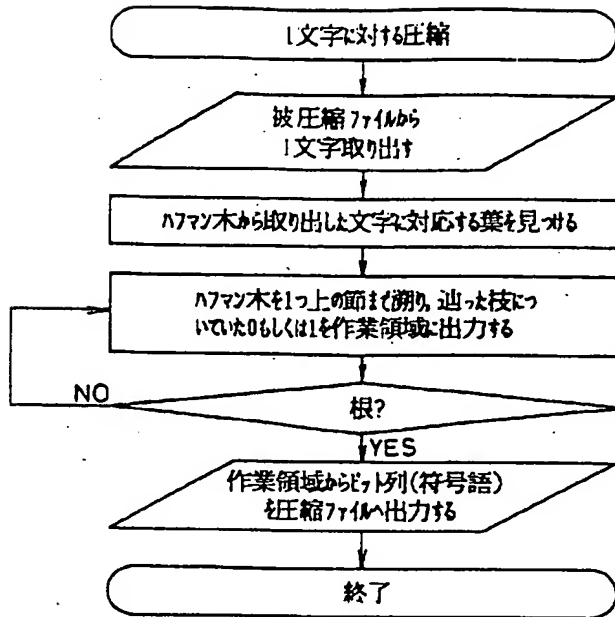
【図3】



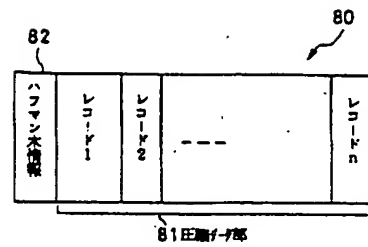
A: 0(12,1)  
B: 100(22,1)  
C: 101(22,1)  
D: 110(22,1)  
E: 111(22,1)

"ADCBABABADC": 0110101010001111000110101(252,444)

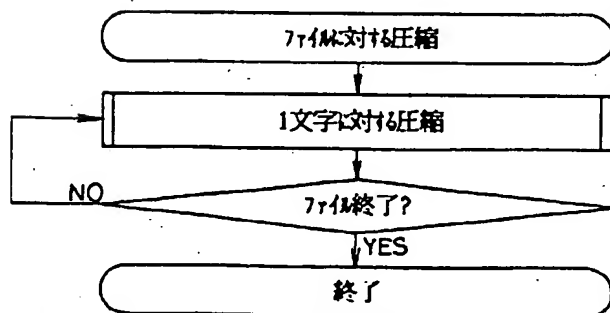
【図4】



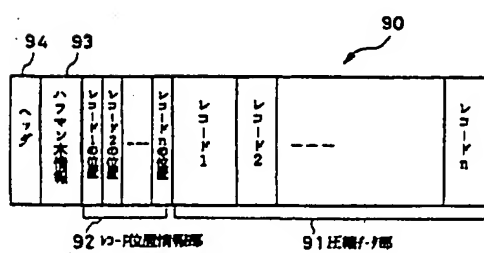
【図9】



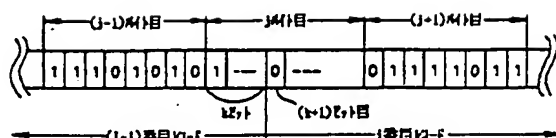
【図5】



【図10】

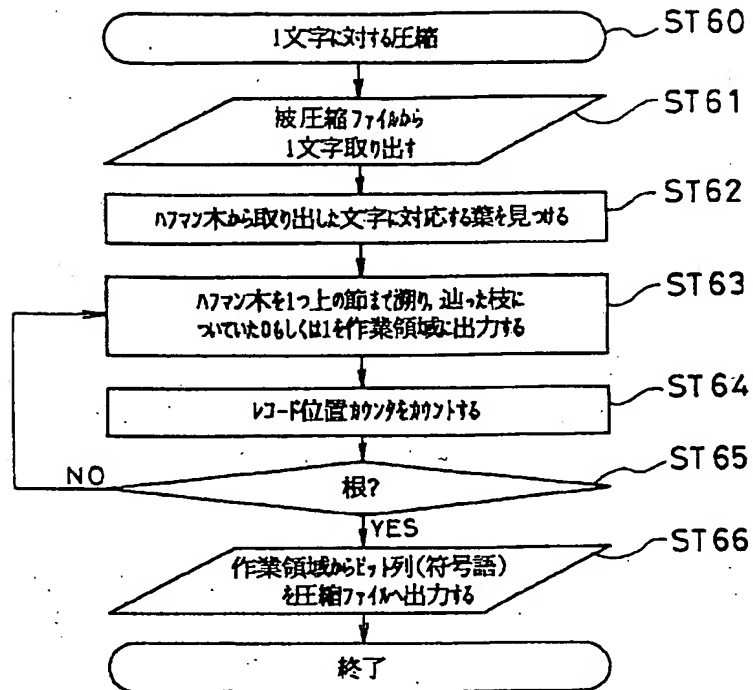


【図11】

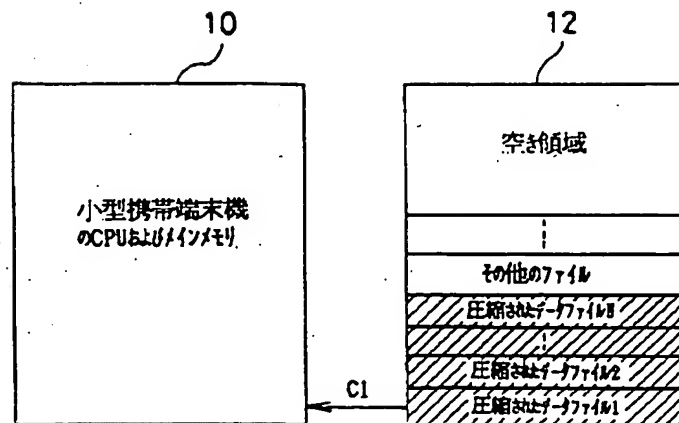




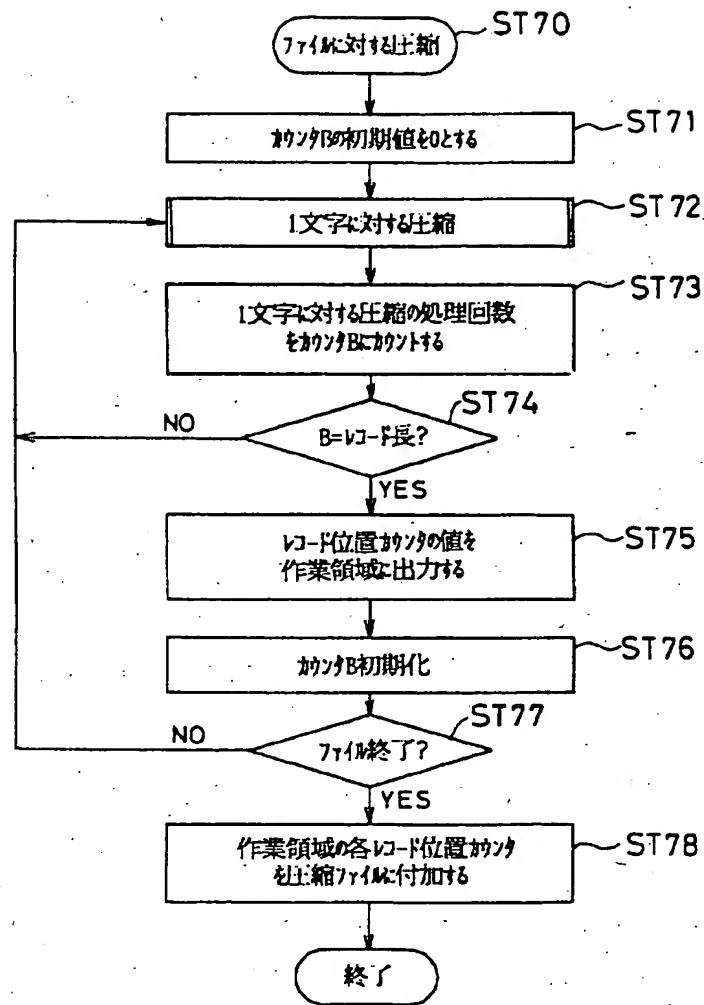
【図6】



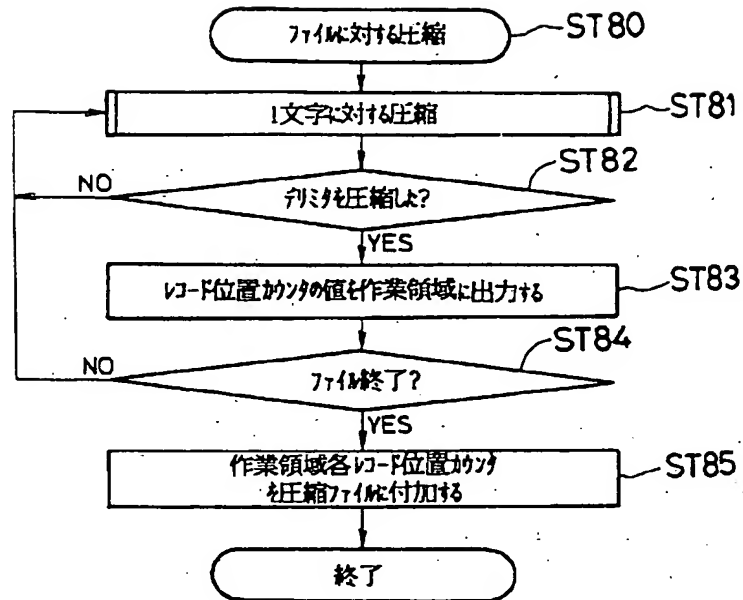
【图 15】



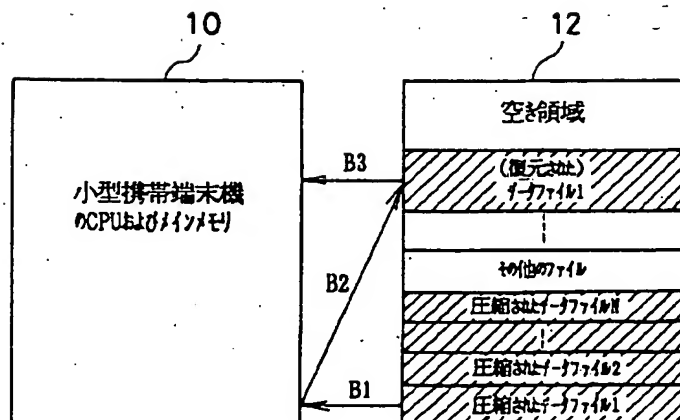
【図7】



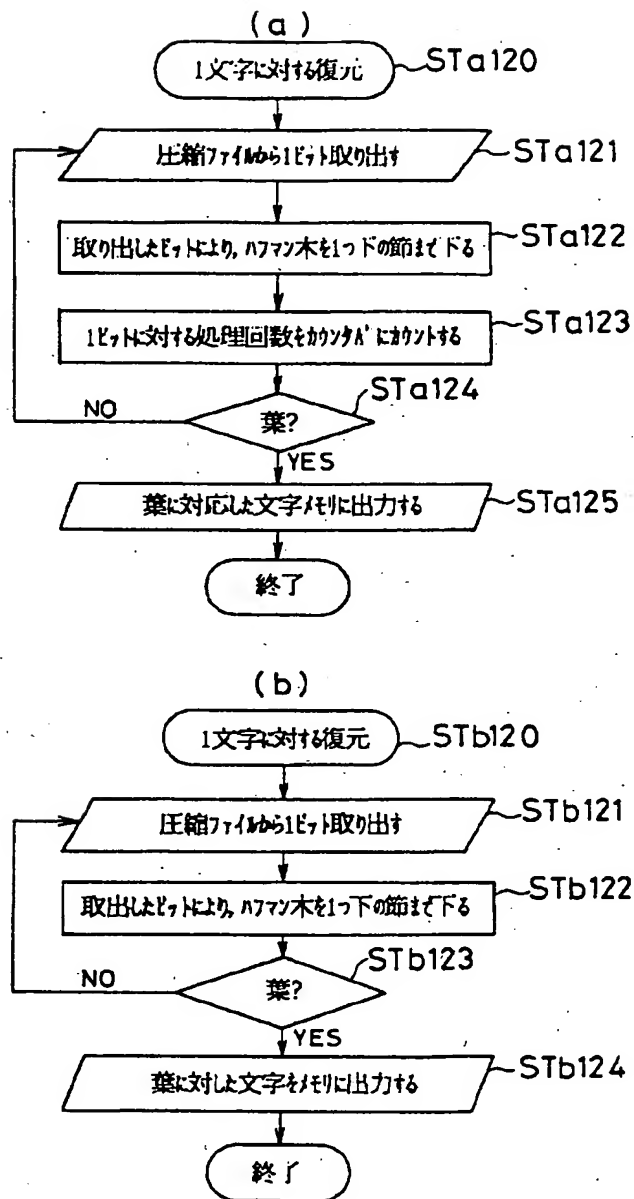
【図8】



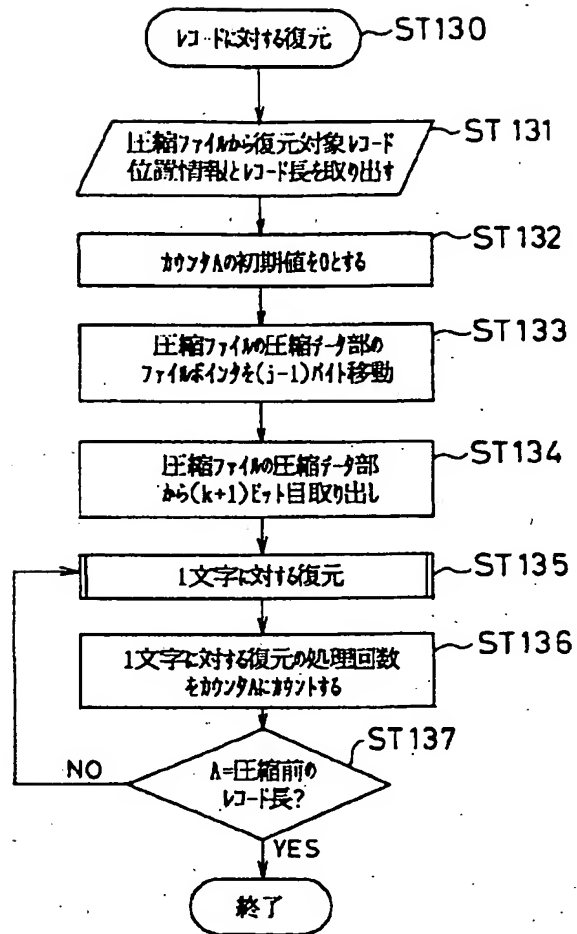
【図16】



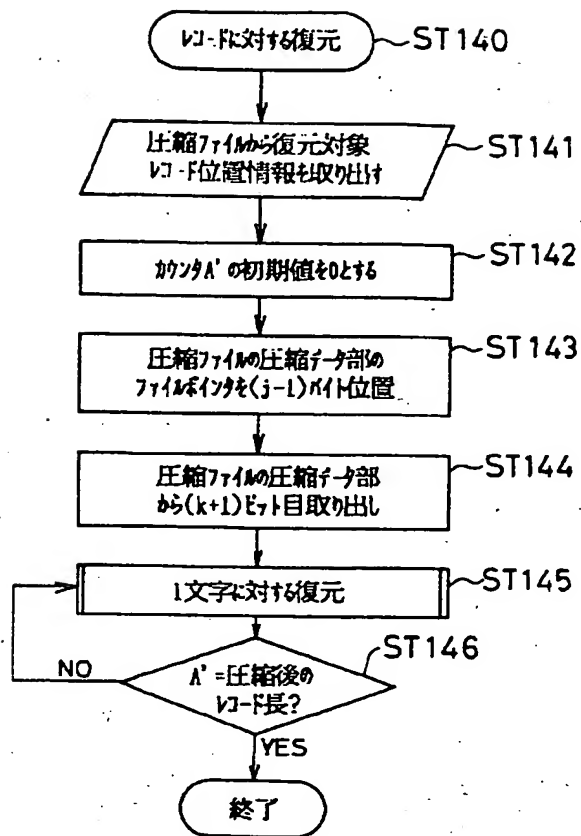
【図12】



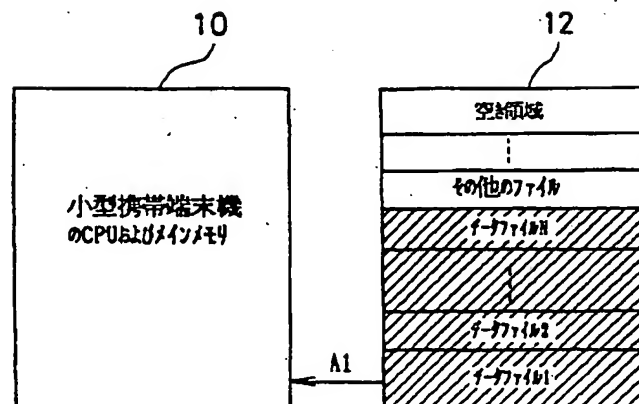
【図13】



【図14】



【図17】



【図18】

